BIGdsPIC6™

User manual

All MikroElektronika´s development systems represent irreplaceable tools for programming and developing microcontroller-based devices. Carefully chosen components and the use of machines of the last generation for mounting and testing thereof are the best guarantee of high reliability of our devices. Due to simple design, a large number of add-on modules and ready to use examples, all our users, regardless of their experience, have the possibility to develop their project in a fast and efficient way.

Development System

MikroElektronika

SOFTWARE AND HARDWARE SOLUTIONS FOR EMBEDDED WORLD ...making it simple

TO OUR VALUED CUSTOMERS

I want to express my thanks to you for being interested in our products and for having confidence in mikroElektronika.
The primary aim of our company is to design and produce high quality electronic products and to constantly improve the performance thereof in order to better suit your needs.

Nebojsa Matic
General Manager

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KeeLoq, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A and other countries.

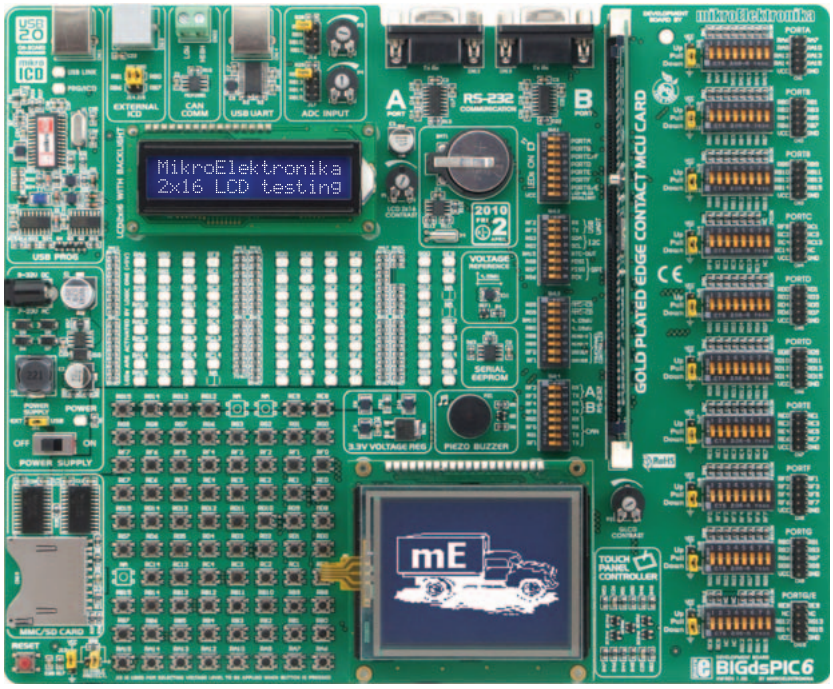
TABLE OF CONTENTS

Introduction to BIGdsPIC6 Development System 4
Key Features 5
1.0. Connecting the System to a PC 6
2.0. Supported Microcontrollers 7
3.0. On-board dsPICFlash Programmer 9
4.0. ICD Connector 10
5.0. Power Supply 11
6.0. A/D Converter Test Inputs 12
7.0. Voltage Reference Source 13
8.0. Serial EEPROM Module 13
9.0. MMC/SD Connector 14
10.0. RS-232 Communication Module 15
11.0. CAN Communication Module 16
12.0. Real-Time Clock (RTC) 17
13.0. USB UART Module 18
14.0. Piezo Buzzer 19
15.0. LEDs 20
16.0. Push Buttons 21
17.0. 2x16 LCD 22
18.0. 128x64 Graphic LCD 23
19.0. Touch Panel 24
20.0. I/O Ports 25

page

Introduction to BIGdsPIC6 Development System

The *BIGdsPIC6*™ development system provides a development environment for programming and experimenting with dsPIC® microcontrollers from Microchip®. The system includes an on-board programmer with mikroICD providing an interface between the microcontroller and a PC. You are simply expected to write a program in one of the dsPIC compilers, generate a .hex file and program your microcontroller using the on-board *dsPICflash*™ programmer. Numerous modules, such as 128x64 graphic LCD, alphanumeric 2x16 LCD, real-time clock etc, are provided on the board and allow you to easily simulate the operation of the target device.



dsPIC DEVELOPMENT BOARD — Full-featured development system for dsPIC microcontroller based devices

USB 2.0 IN-CIRCUIT PROGRAMMER — USB 2.0 on-board programmer

mikro ICD IN-CIRCUIT DEBUGGER — Built-in debugger for testing programs in real time at hardware level

MMC/SD MEMORY CARD — A possibility of reading MMC/SD memory cards

GLCD128x64 — Graphic LCD with backlight

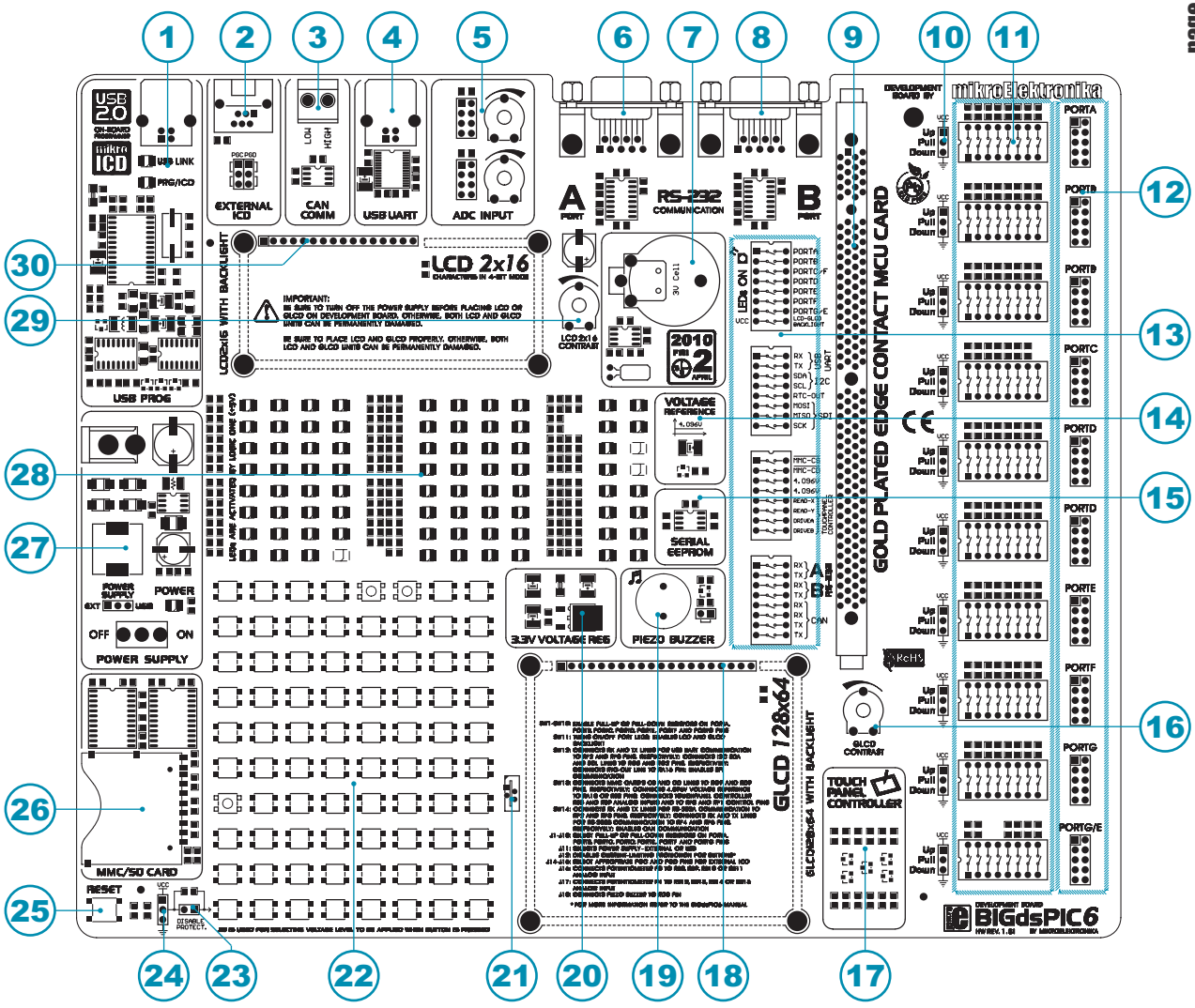
The *dsPICFLASH* program provides a complete list of all supported microcontrollers. The latest version of this program with updated list of supported microcontrollers can be downloaded from our website at **www.mikroe.com**

Package contains:

Development system:	BIGdsPIC6
CD:	product CD with relevant software
Cables:	USB cable
Documentation:	Manuals for BIGdsPIC6 and dsPICflash, quick guide for installing USB drivers, electrical schematic of the system

System specification:

Power Supply:	over an AC/DC connector (7-23V AC or 9-32V DC); or over a USB cable for programming (5V DC)
Power consumption:	~40mA in idle state when all on-board modules are off
Dimension:	26,5 x 22cm (10,4 x 8,6inch)
Weight:	462g (0.89lbs)



Key Features

1. USB 2.0 programmer with *mikroICD* support
2. Connector for Microchip's debugger (ICD2 or ICD3)
3. CAN communication module
4. USB UART Module
5. A/D converter test inputs
6. Connector A for RS-232 communication
7. Real-Time Clock
8. Connector B for RS-232 communication
9. DIMM-168P socket for MCU cards
10. Jumper for pull-up/pull-down resistor selection
11. DIP switches enable pull-up/pull-down resistors
12. I/O port connectors
13. DIP switches turn on/off on-board modules
14. 4.096V voltage reference
15. Serial EEPROM module
16. Potentiometer for adjusting graphic display contrast
17. Touch panel controller
18. Graphic LCD connector
19. Piezo buzzer
20. 3.3V voltage regulator
21. Touch panel connector
22. Push buttons simulate microcontroller digital inputs
23. Jumper for protective resistor shortening
24. Jumper for selecting push buttons' logic state
25. Reset button
26. MMC/SD card connector
27. Power supply voltage regulator
28. 77 LEDs indicate pins' logic state
29. Potentiometer for adjusting LCD contrast
30. Alphanumeric LCD display connector

1.0. Connecting the System to a PC

Step 1:

Follow the instructions provided in the relevant manuals and install the *dsPICFLASH* program and USB driver from the product CD. USB drivers are essential for the proper operation of the on-board programmer.
In case you already have one of the Mikroelektronika's dsPIC compilers installed on your PC, there is no need to reinstall USB drivers as they are already installed along with the compiler.

Step 2:

Use the USB cable to connect the development system to a PC. One end of the USB cable, with a USB connector of B type, should be connected to the development system, as shown in Figure 1-2, whereas the other end of the cable with a USB connector of A type should be connected to a PC. When establishing a connection, make sure that jumper J11 is placed in the USB position as shown in Figure 1-1.

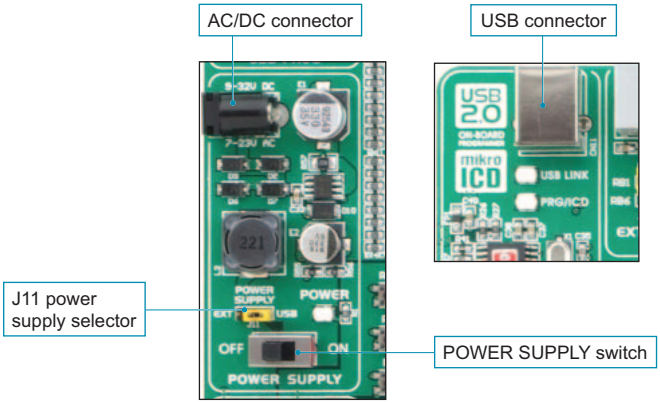


Figure 1-1: Power supply

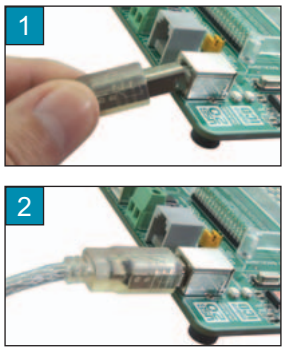


Figure 1-2: Connecting USB cable

Step 3:

Turn on your development system by setting the POWER SUPPLY switch to the ON position. Two LEDs marked as POWER and USB LINK will be automatically turned on indicating that your development system is ready to use. Use the on-board programmer and the *dsPICFLASH* program to dump a hex code into the microcontroller and employ the system to test and develop your projects.

NOTE: If some additional modules are used, such as LCD, GLCD etc, it is necessary to place them properly on the development system while it is turned off. Otherwise, either can be permanently damaged. Refer to figure below for the proper placing of the additional modules.

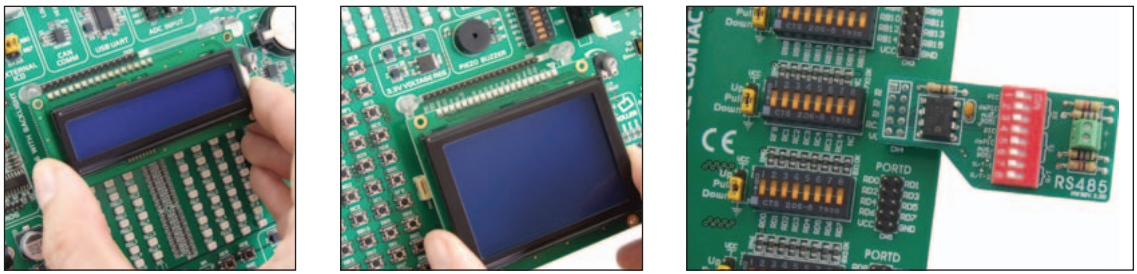


Figure 1-3: Placing additional modules on the board

2.0. Supported Microcontrollers

The *BIGdsPIC6* development system provides a DIMM-168P connector to place an MCU card into. This development system comes with an MCU card with a microcontroller in 80-pin TQFP package soldered on it, Figure 2-3. Besides, the MCU card alone provides an oscillator as well as 80 soldering pads connected to the microcontroller pins. Each pad is marked the same as the pin it is connected to. Soldering pads also make connection between the MCU card and target device's modules easy.

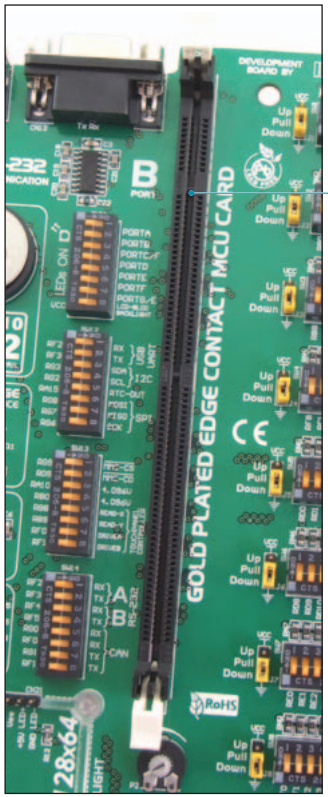


Figure 2-1: DIMM-168P connector

DIMM-168P connector for placing the MCU card with a microcontroller in TQFP package

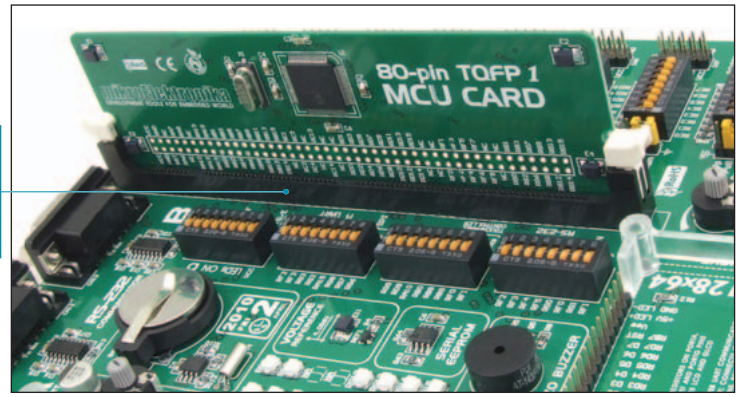


Figure 2-2: DIMM-168P connector with the MCU card inserted

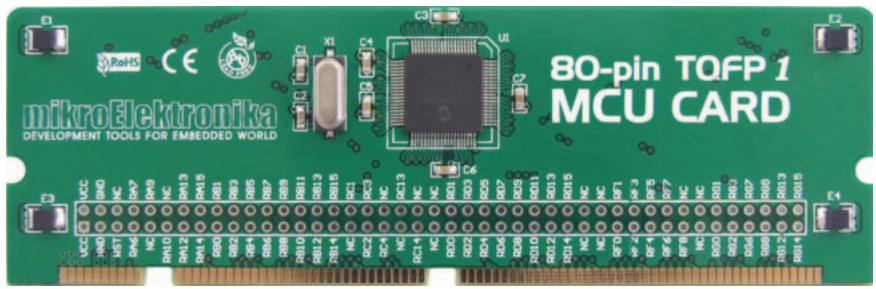


Figure 2-3: MCU card with an 80-pin microcontroller in TQFP package soldered on it

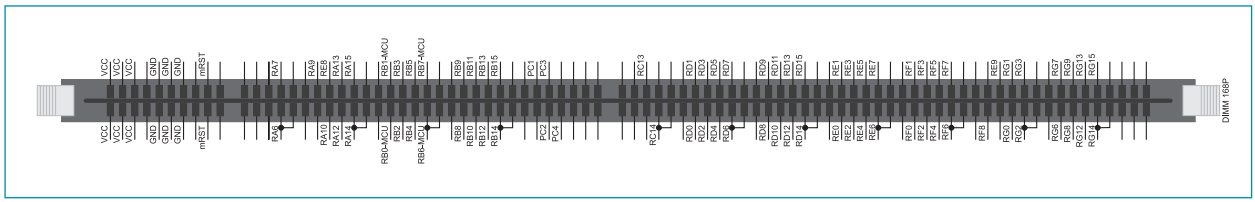
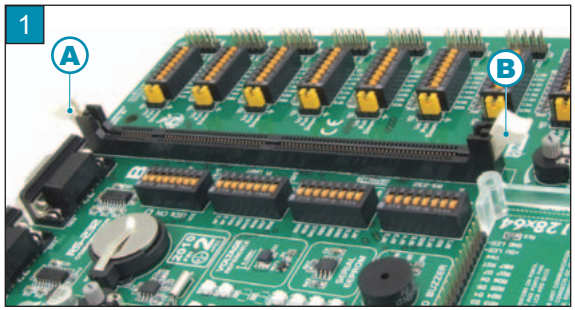
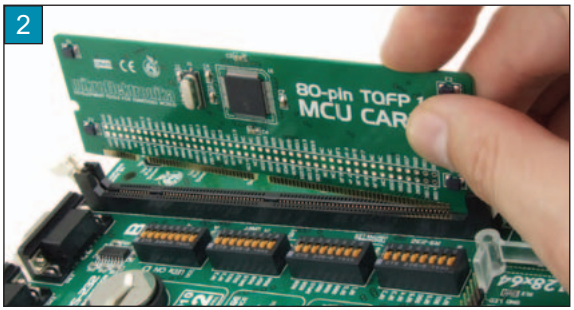


Figure 2-4: Schematic of the DIMM-168P connector's pinout

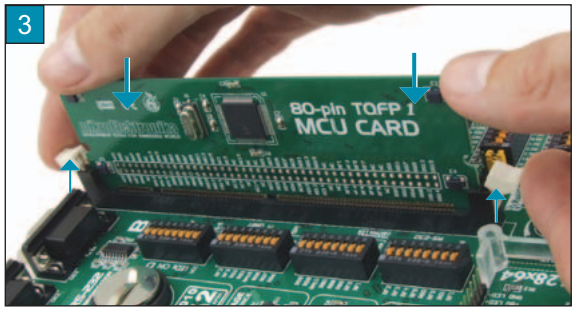
Placing MCU card into the DIMM-168P connector is performed as follows:



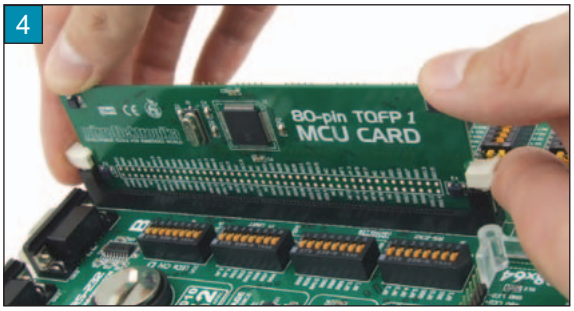
Open extraction levers A and B



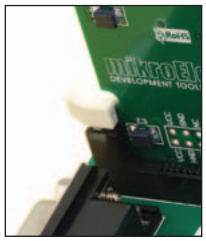
Place the MCU card into the DIMM-168P connector

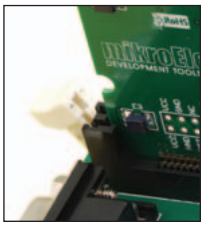


Push the MCU card down gently into the DIMM-168P connector and lift extraction levers slowly at the same time



Close the extraction levers when the MCU card is properly placed into the connector


Extraction levers used to fix the MCU card in 'closed' position


Extraction levers used to fix the MCU card in 'open' position

3.0. On-board dsPICFlash Programmer

A programmer is a necessary tool when working with microcontrollers. It is used to load a hex code into the microcontroller and provides an interface between the microcontroller and a PC. The *BIGdsPIC6* features an on-board *dsPICFlash* programmer. The *dsPICFLASH* program is used for loading a .hex file into the microcontroller. Figure 3-3 shows connection between the compiler, *dsPICFLASH* program and the microcontroller.

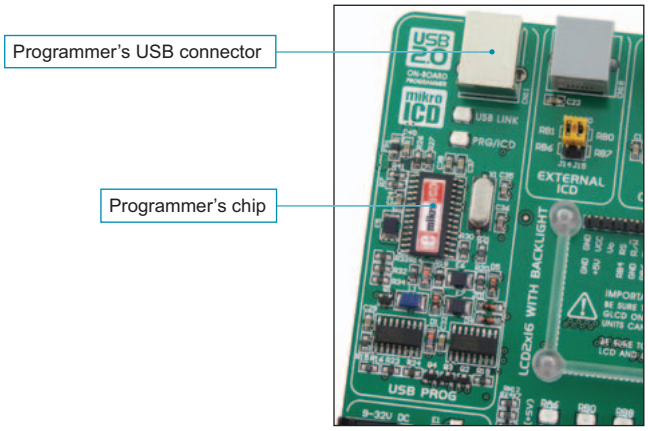


Figure 3-1: On-board programmer

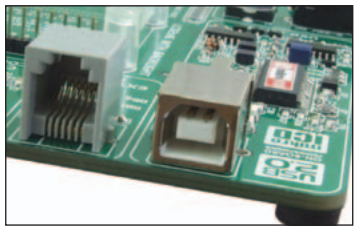


Figure 3-2: USB connector's front side

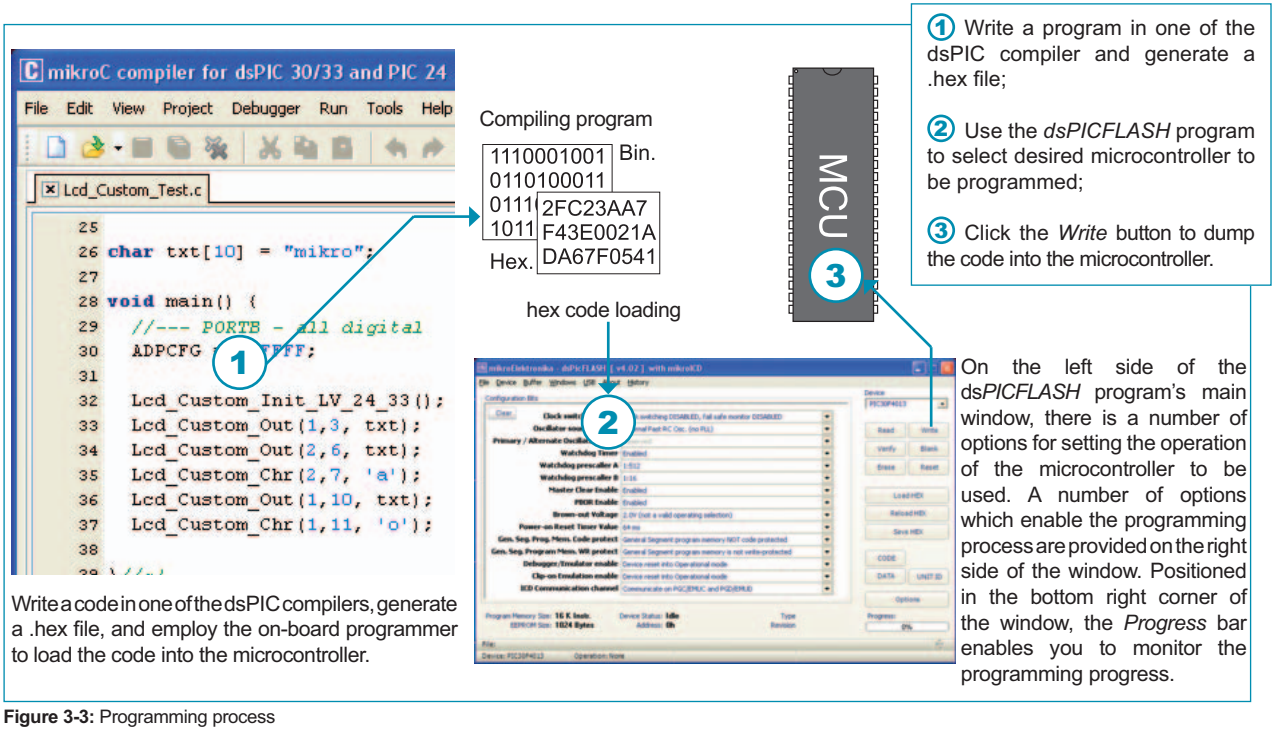

mikroC compiler for dsPIC 30/33 and PIC 24
File Edit View Project Debugger Run Tools Help
Lcd_Custom_Test.c
char txt[10] = "mikro";
void main() {
//--- PORTB - all digital
Lcd_Custom_Init_LV_24_33();
Lcd_Custom_Out(1,3, txt);
Lcd_Custom_Out(2,6, txt);
Lcd_Custom_Chr(2,7, 'a');
Lcd_Custom_Out(1,10, txt);
Lcd_Custom_Chr(1,11, 'o');
1
Compiling program
1110001001
0110100011
0111
1011
Bin.
2FC23AA7
F43E0021A
DA67F0541
Hex.
hex code loading
2
MCU
3
① Write a program in one of the dsPIC compiler and generate a .hex file;
② Use the dsPICFLASH program to select desired microcontroller to be programmed;
③ Click the Write button to dump the code into the microcontroller.
Write a code in one of the dsPIC compilers, generate a .hex file, and employ the on-board programmer to load the code into the microcontroller.
On the left side of the dsPICFLASH program's main window, there is a number of options for setting the operation of the microcontroller to be used. A number of options which enable the programming process are provided on the right side of the window. Positioned in the bottom right corner of the window, the Progress bar enables you to monitor the programming progress.

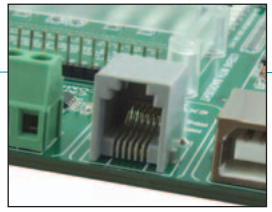
Figure 3-3: Programming process

page

4.0. ICD Connector

The ICD connector enables communication between the microcontroller and an external ICD debugger/programmer (ICD2 or ICD3) from Microchip. Jumpers J14 and J15 are used for selecting the pin to be fed with programming signal. For the programming signal to be sent to the microcontroller, it is necessary to place both jumpers in the same position, as shown in Figure 4-3. Either RB0 and RB1 or RB6 and RB7 microcontroller pins can be fed with this signal. The position of jumpers J14 and J15 depends on the microcontroller provided on the MCU card. When the dsPIC30F6014 microcontroller is programmed by means of the external ICD programmer, jumpers J14 and J15 should be placed in the RB0 and RB1 position, respectively. In case some other microcontroller is used, it is necessary to check which microcontroller pins can be fed with the programming signal.

Figure 4-1: Pins RB6 and RB7 are used for programming



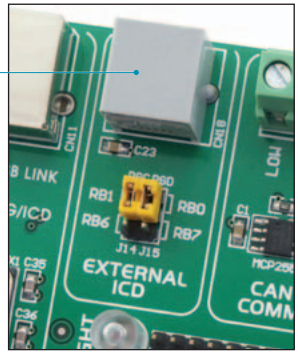


Figure 4-2: Pins RB0 i RB1 are used for programming

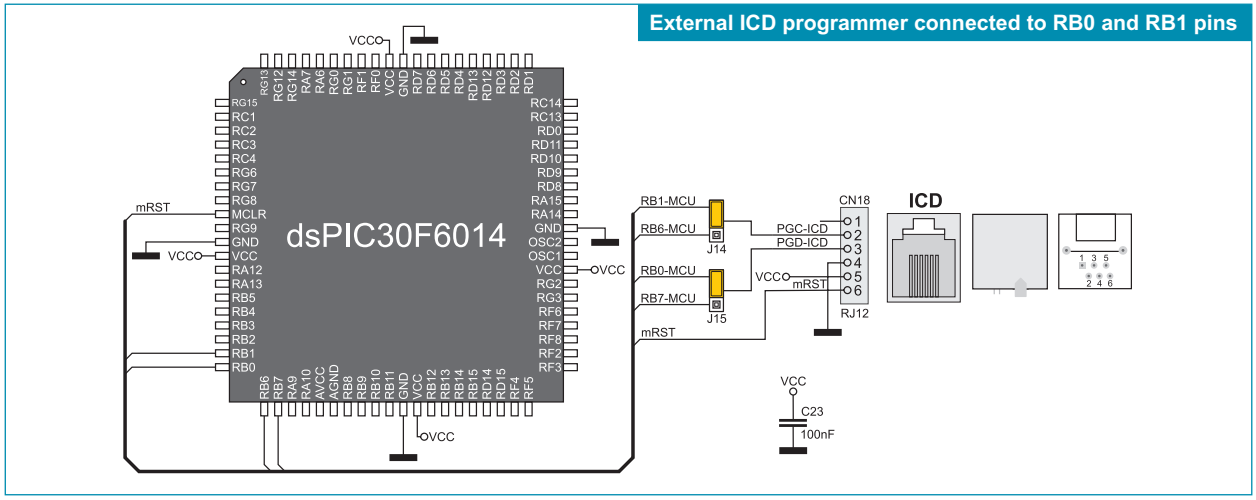


Figure 4-3: ICD connector's pinout and pin designations

5.0. Power supply

The *BIGdsPIC6* development system may use one of two power supply sources:

1. +5V PC power supply through the USB programming cable; and
2. External power supply source connected to an AC/DC connector provided on the development board.

The MC34063A voltage regulator and Gretz rectifier are used to enable external power supply voltage to be either AC (in the range of 7V to 23V) or DC (in the range of 9V to 32V). Jumper J11 is used as a power supply source selector. To make advantage of the USB power supply, jumper J11 should be placed in the USB position. When using the external power supply, jumper J11 should be placed in the EXT position. The development system is turned on/off by switching the position of the POWER SUPPLY switch.

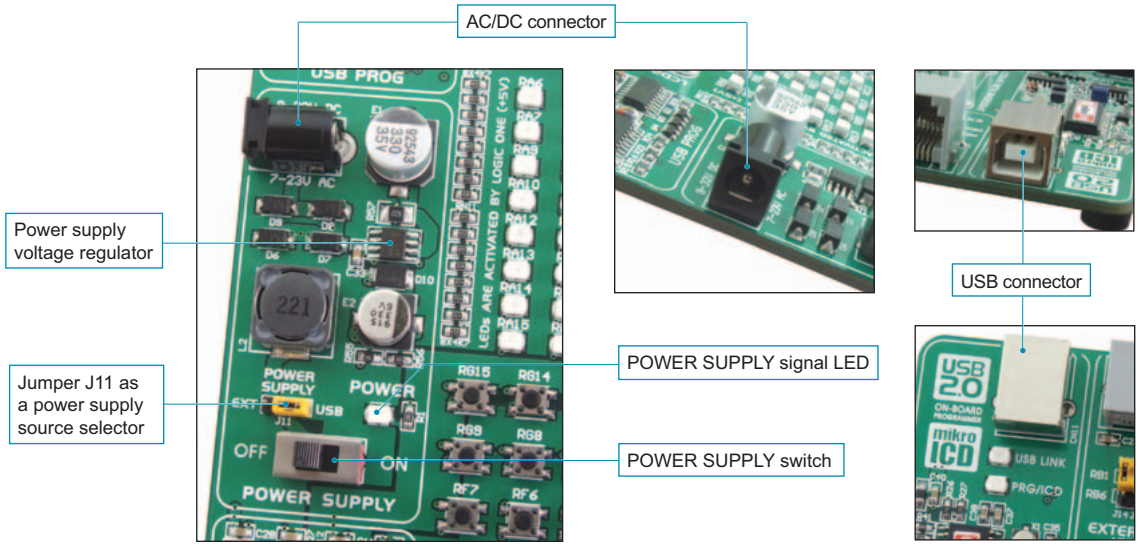


Figure 5-1: Power supply

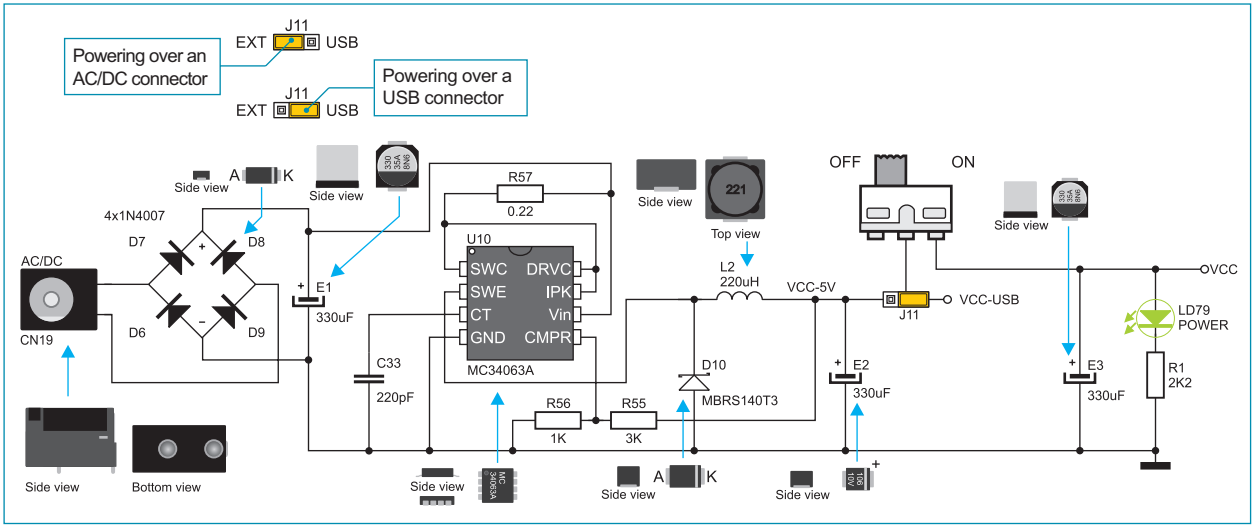


Figure 5-2: Power supply source schematic

page

6.0. A/D Converter Test Inputs

An A/D converter is used for converting an analog voltage into the appropriate digital value. The A/D converter is linear, which means that converted number is linearly dependent on the input voltage value. The A/D converter built into the microcontroller converts an analog voltage value into a 10-bit number. Potentiometers P3 and P4 enable voltage to vary between 0 and 5V. The microcontroller with a built-in A/D converter is supplied with this voltage via test inputs. Jumpers J16 and J17 are used for selecting one of the following pins RB8 - RB15 to be supplied with A/D conversion voltage. Resistors R28 and R29 have a protective function and are used to limit current flow through the potentiometer or the microcontroller pin.

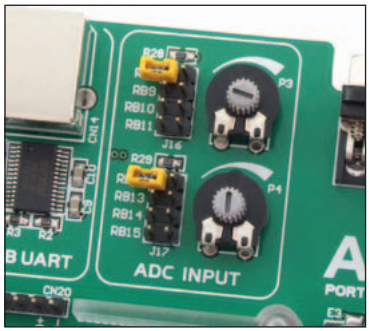


Figure 6-1: ADC (jumpers in default position)

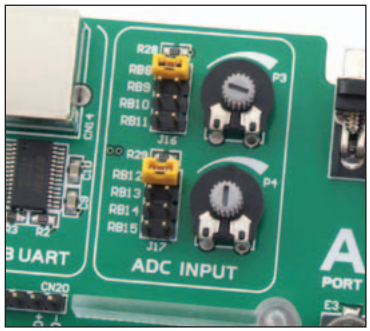


Figure 6-2: Pins RB8 and RB12 used as input pins for A/D conversion

A/D conversion is performed via RB8 and RB12 microcontroller pins

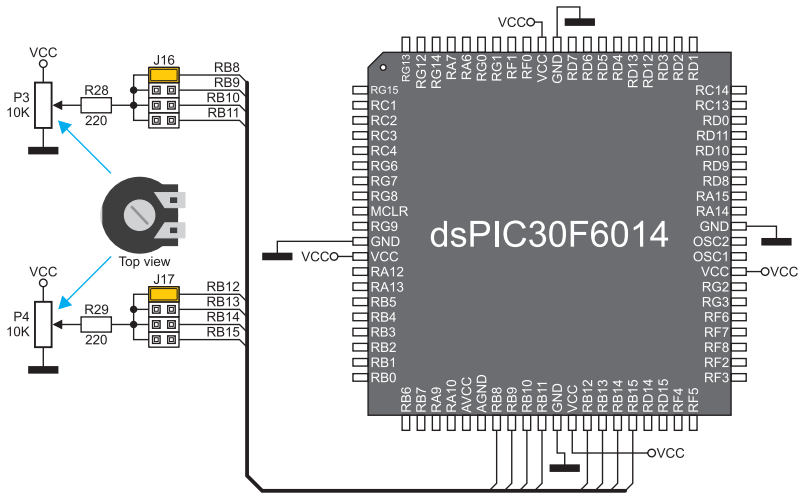


Figure 6-3: A/D converter test inputs and microcontroller connection schematic

NOTE: In order to enable the microcontroller to accurately perform A/D conversion, it is necessary to turn off LEDs and pull-up/pull-down resistors on the port pins used by the A/D converter.

7.0. Votage Reference Source

BIGdsPIC6 provides the MCP1541 circuit which generates the voltage reference used for A/D conversion. The value of the voltage reference is 4.096V and it is brought to the microcontroller via RA10 or RB0 pin.

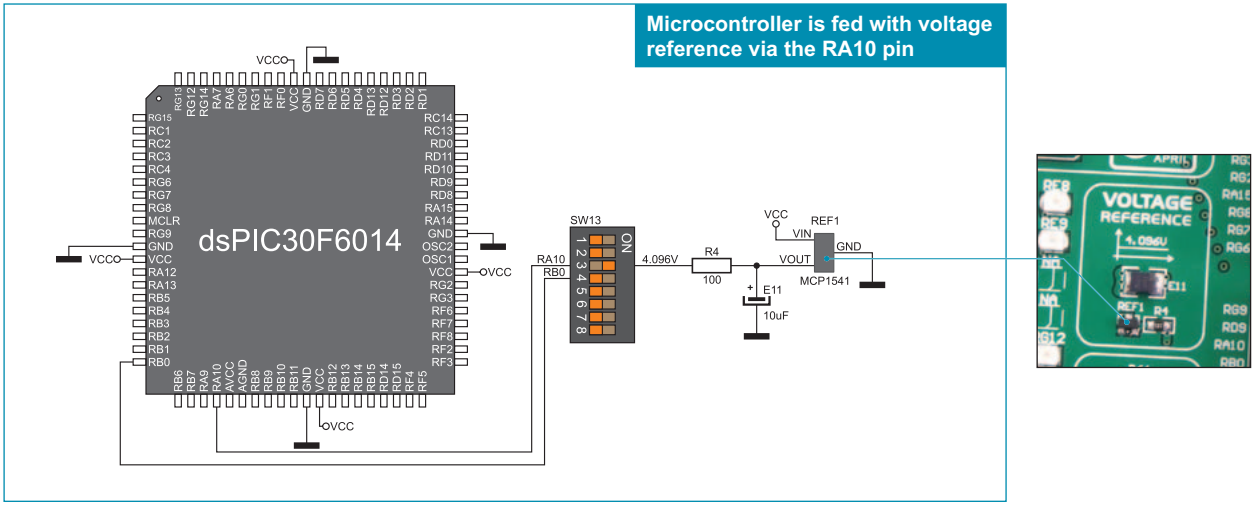


Figure 7-1: Voltage reference connection schematic

8.0. Serial EEPROM Module

EEPROM (Electrically Erasable Programmable Read-Only Memory) is a built-in memory module used to store data that should be saved when the power supply goes off. The 24AA01 circuit can store up to 1Kbit data and communicates with the microcontroller through pins RG2 and RG3 using I^2C communication. In order to enable connection between the EEPROM module and the microcontroller, it is necessay to set switches 3 and 4 on the DIP switch SW12 to the ON the position.

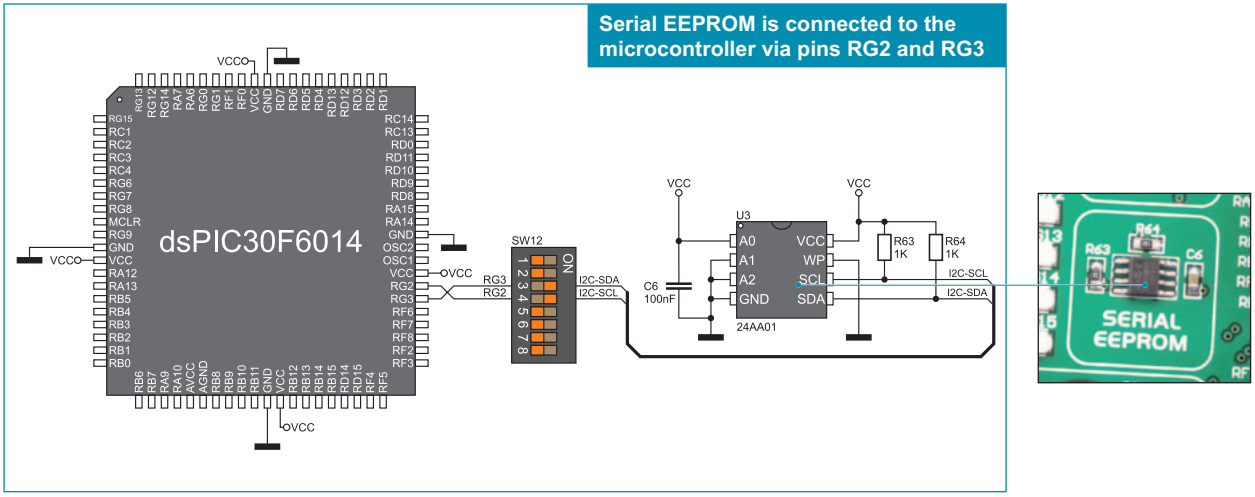


Figure 8-1: Serial EEPROM module and microcontroller connection schematic

page

9.0. MMC/SD Connector

The MMC/SD connector enables the memory card to be interfaced to the microcontroller in order to expand microcontroller memory. To enable serial communication between the microcontroller and the memory card it is necessary to adjust their voltage levels. Memory card is powered with the 3.3V power supply voltage (VCC-MMC) generated by the REG1 voltage regulator, whereas the value of the microcontroller power suplly voltage is 5V (VCC). A bus transceiver 74LVCC3245 is used here to adjust these voltage levels. In addition, in order to establish communication between memory card and microcontroller, it is necessary to set switches 6, 7 and 8 on the DIP switch SW12 as well as switches 1 and 2 on the DIP switch SW13 to the ON position.

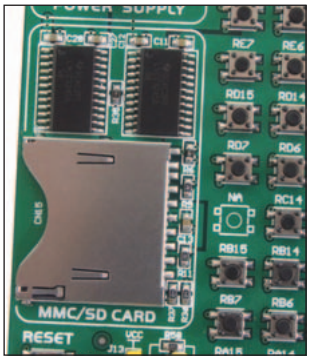


Figure 9-1: MMC/SD connector

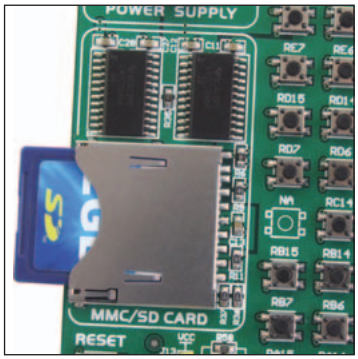


Figure 9-2: MMC/SD memory card

MMC/SD card communicates to the microcontroller via MISO, MOSI, SCK, MMC-CS and MMC-CD lines

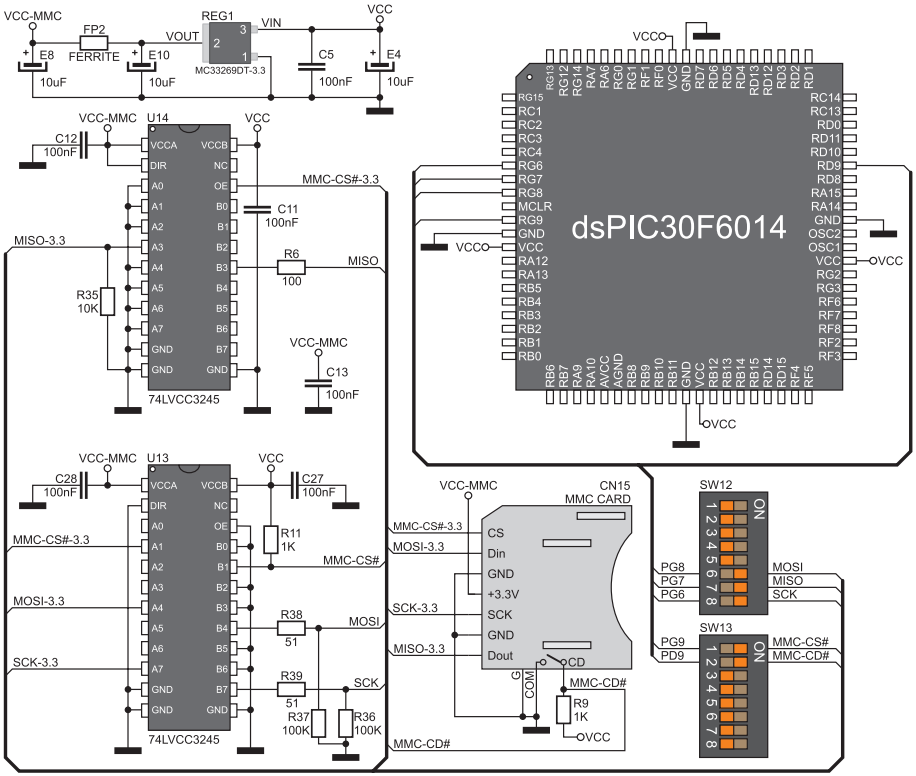


Figure 9-3: MMC/SD connector and microcontroller connection schematic

10.0. RS-232 Communication Module

USART (*Universal Synchronous/Asynchronous Receiver/Transmitter*) is one of the most common ways of exchanging data between the PC and peripheral units. RS-232 serial communication is performed through a 9-pin SUB-D connector and the microcontroller USART module. The *BIGdsPIC6* provides two RS-232 ports, RS-232A and RS-232B. Use switches RX232-A and TX232-A on the DIP switch SW14 to enable port RS-232A. Likewise, use switches RX232-B and TX232-B on the DIP switch SW14 to enable port RS-232B. The microcontroller pins used in such communication are marked as follows: RX - *receive data line* and TX - *transmit data line*. Data rate goes up to 115 kbps.

In order to enable the USART module of the microcontroller to receive input signals with different voltage levels, it is necessary to adjust these levels using an IC circuit such as MAX202 (MAX232).

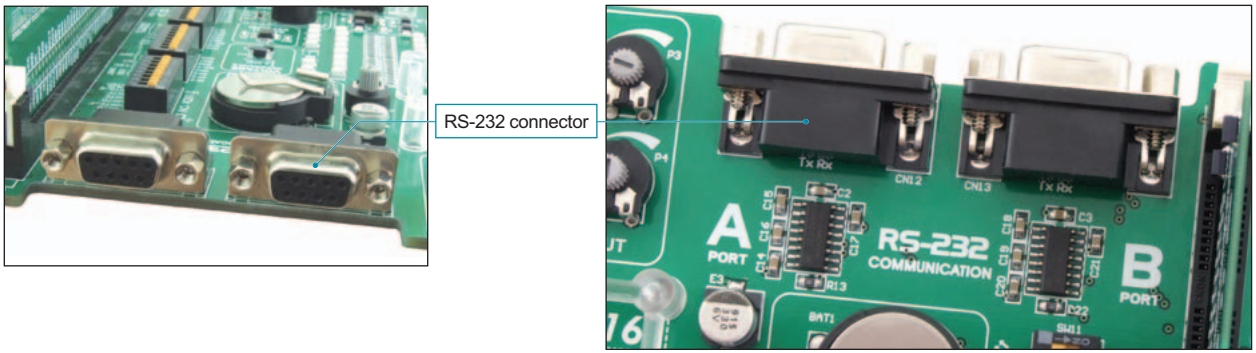


Figure 10-1: RS-232 module

The function of switches 1, 2, 3 and 4 on the DIP switch SW14 is to determine which of the microcontroller pins are to be used as RX and TX lines, Figure 10-2.

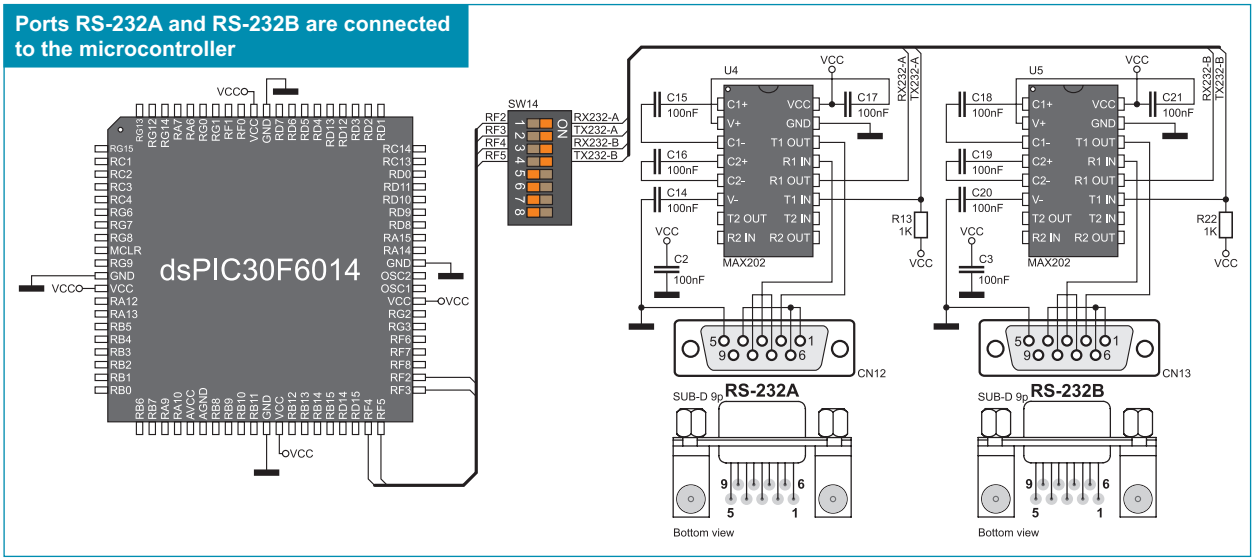


Figure 10-2: RS-232 module connection schematic

NOTE: Make sure that your microcontroller is provided with the USART module as it is not necessarily integrated in all dsPIC microcontrollers.

page

11.0. CAN Communication Module

CAN (Controller Area Network) is a communication standard primarily intended for use in automotive industry. It enables the microcontroller to communicate to a car device without using a host PC. In addition, such communication is widely used in industrial automation. The *BIGdsPIC6* uses the MCP2551 circuit for CAN communication. This circuit provides an interface between the microcontroller and some peripheral device. To enable connection between the microcontroller and MCP2551, it is necessary to set switches 5 and 7 on the DIP switch SW14 to the ON position. Switches 6 and 8 on the DIP switch SW14 may also be used for this purpose. Which of these switches is to be used depends solely on the arrangement of the microcontroller pins connected to CAN communication lines. The RX line is connected to the microcontroller via RG0 and RF0 microcontroller pins, whereas the TX line is connected to the microcontroller via RG1 and RF1 microcontroller pins.

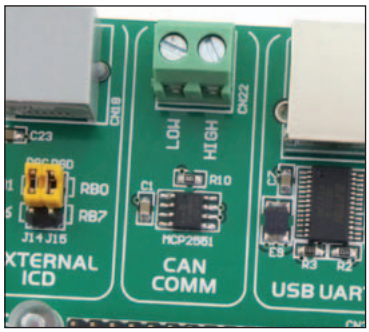


Figure 11-1: CAN module

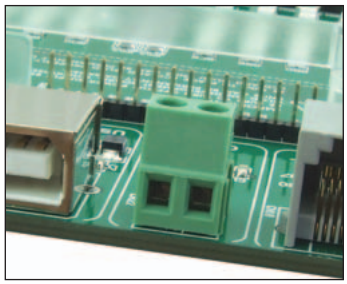


Figure 11-2: CAN module connector

CAN communication is enabled via DIP switch SW14

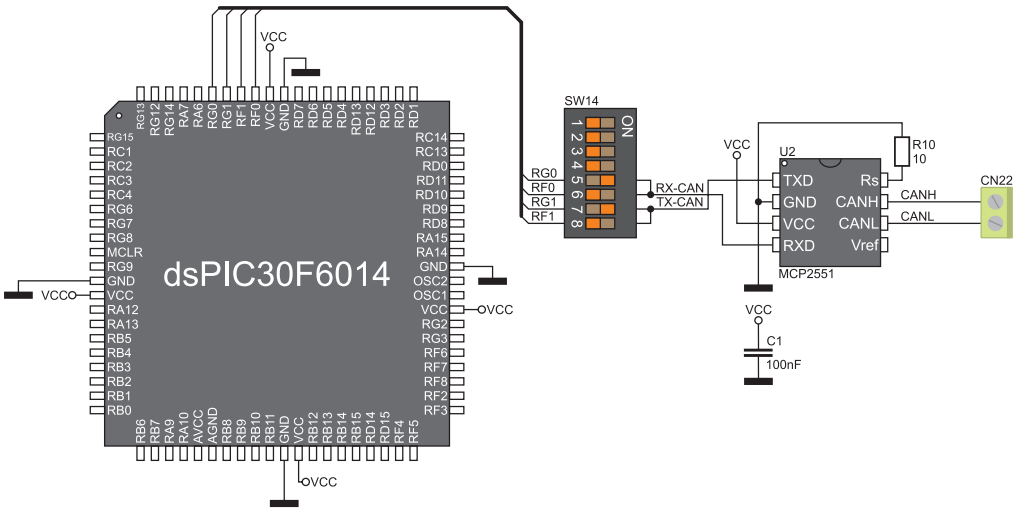


Figure 11-3: Microcontroller and MCP2551 connection schematic

12.0. Real-Time Clock (RTC)

The DS1307 circuit enables the *BIGdsPIC6* development system to keep the real time. The real-time clock's main features are as follows:

- providing information on seconds, minutes, hours, days, days in a week and dates including corrections for a leap year
- I^2C serial interface
- automatic power-fail detection
- power consumption less than 500nA

The real-time clock is widely used in alarm devices, industrial controllers, mass-consumption products etc. The real-time clock provided on the *BIGdsPIC6* development system is used to generate an interrupt at pre-set time. In order to establish connection between the microcontroller and real-time clock it is necessary to set switches RG3, RG2 and RA15 on the DIP switch SW12 to the ON position.

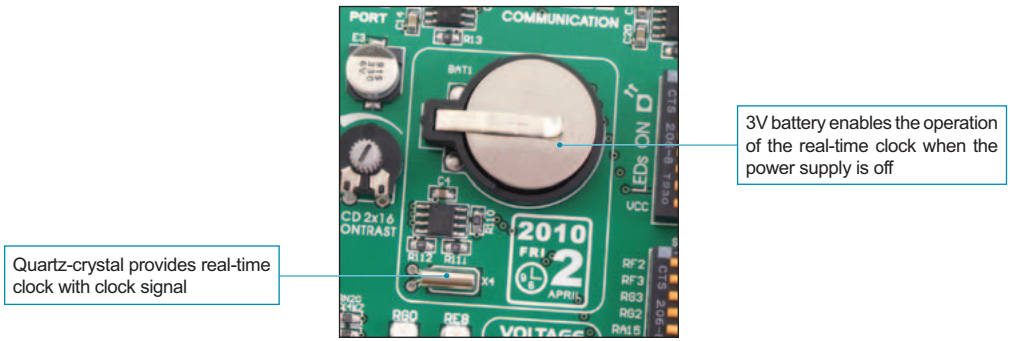

3V battery enables the operation of the real-time clock when the power supply is off
Quartz-crystal provides real-time clock with clock signal

Figure 12-1: Real-time clock

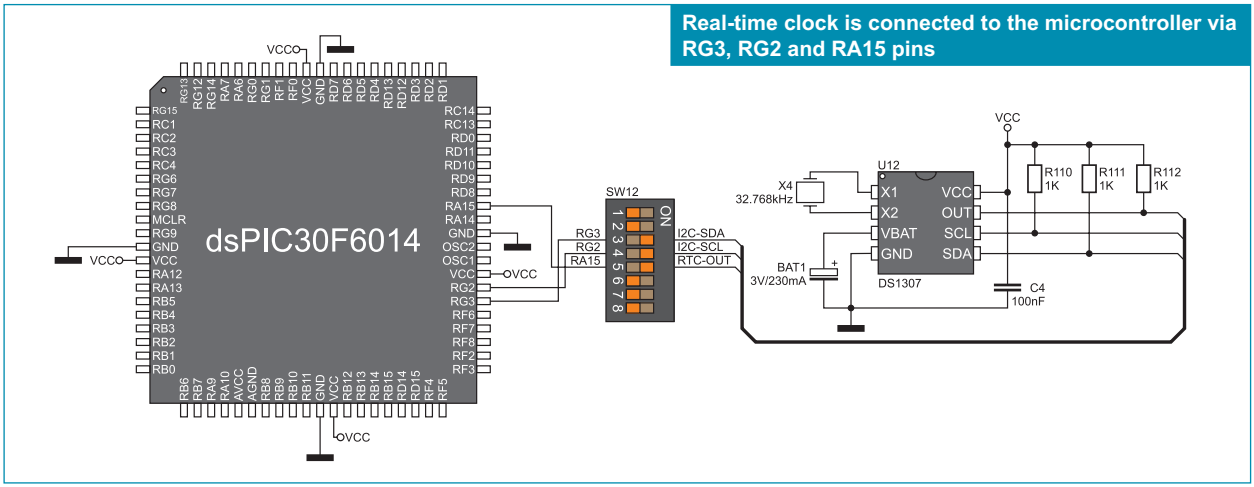


Figure 12-2: Real-time clock and microcontroller connection schematic

page

13.0. USB UART Module

The USB UART module is used to connect the microcontroller provided on the development system to external USB devices. The USB UART module features the FT232RL circuit providing an interface between the USB device and serial UART module integrated into the microcontroller. In order to establish connection between the microcontroller and USB UART module, it is necessary to set switches 1 (RF2) and 2 (RF3) on the DIP switch SW12 to the ON position.

When connecting a UART device and the microcontroller, it is necessary to cross connect data send and data receive lines. In other words, the UART's RX pin should be connected to the microcontroller's TX pin, while the TX pin should be connected to the RX pin. It is not possible to directly connect UART's and microcontroller's TX lines since the microcontroller wouldn't be able to receive nor send data.

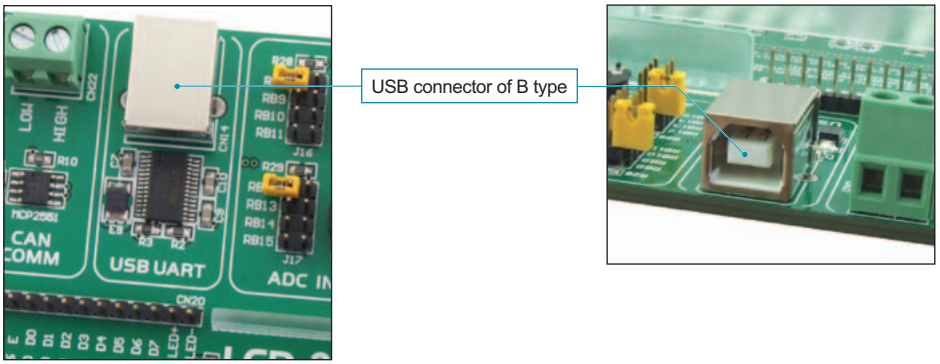


Figure 13-1: USB UART module

USB connector of B type is connected to the microcontroller

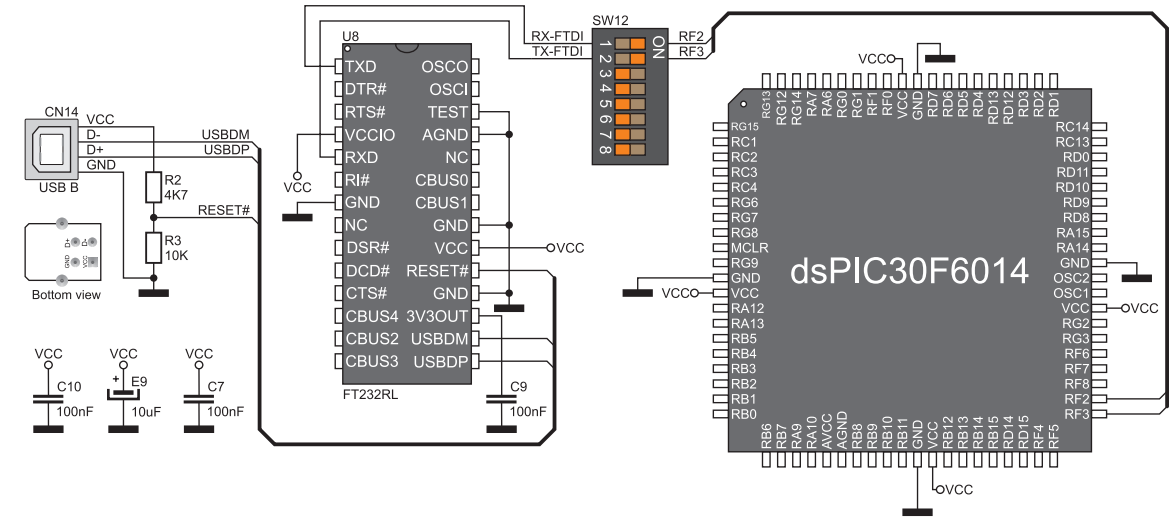


Figure 13-2: USB connector and microcontroller connection schematic

14.0. Piezo Buzzer

Due to a built-in piezo buzzer, the develompent system is capable of emitting audio signals. For the piezo buzzer to operate normally it is necessary to generate a voltage signal of specified frequency. The voltage signal is generated in the microcontroller by the appropriate code written to its memory. Remember, when writing the voltage signal generation code, the piezo buzzer's resonant frequency is 3.8kHz. In addition, other frequencies in the range between 20Hz and 20kHz can be used, but the best performance is provided by frequencies ranging between 2kHz and 4kHz. The voltage signal can be generated via the RD8 microcontroller pin. Jumper J18 is used to connect the piezo buzzer with the RD8 microcontroller pin.

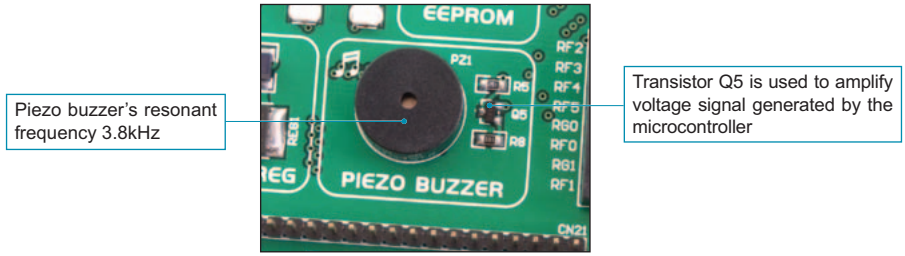


Figure 14-1: Piezo Buzzer

Piezo buzzer is connected to the microcontroller via pin RD8

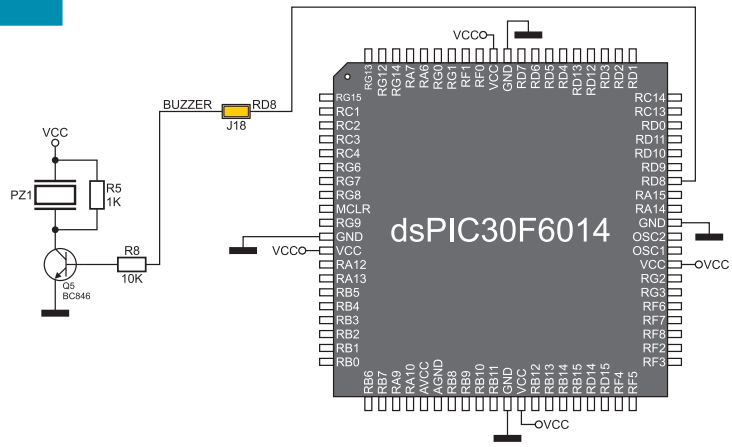


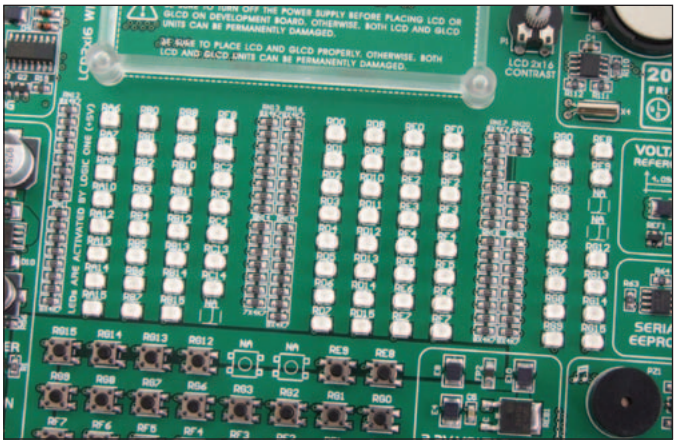
Figure 14-2: Piezo buzzer and microcontroller connection schematic

page

15.0. LEDs

LED (Light-Emitting Diode) is a highly efficient electronic light source. When connecting LEDs, it is necessary to use a current limiting resistor. A common LED voltage is approximately 2.5V, while the current varies from 1 to 20mA depending on the type of LED. The *BIGdsPIC6* uses LEDs with current I=1mA.

There are 77 LEDs on the *BIGdsPIC6* development system which visually indicate the state of each microcontroller I/O pin. An active LED indicates that a logic one (1) is present on the pin. In order to enable the pin state to be shown, it is necessary to select appropriate port (PORTA, PORTB, PORTC, PORTD, PORTE, PORTF, PORTG or PORTG/E) using the DIP switch SW11.



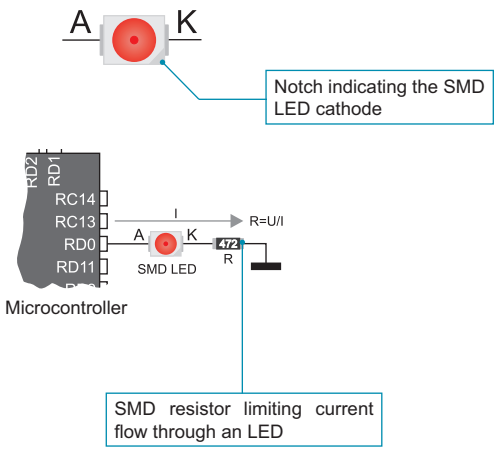


Figure 15-1: LEDs

Port PORTB LEDs are turned on

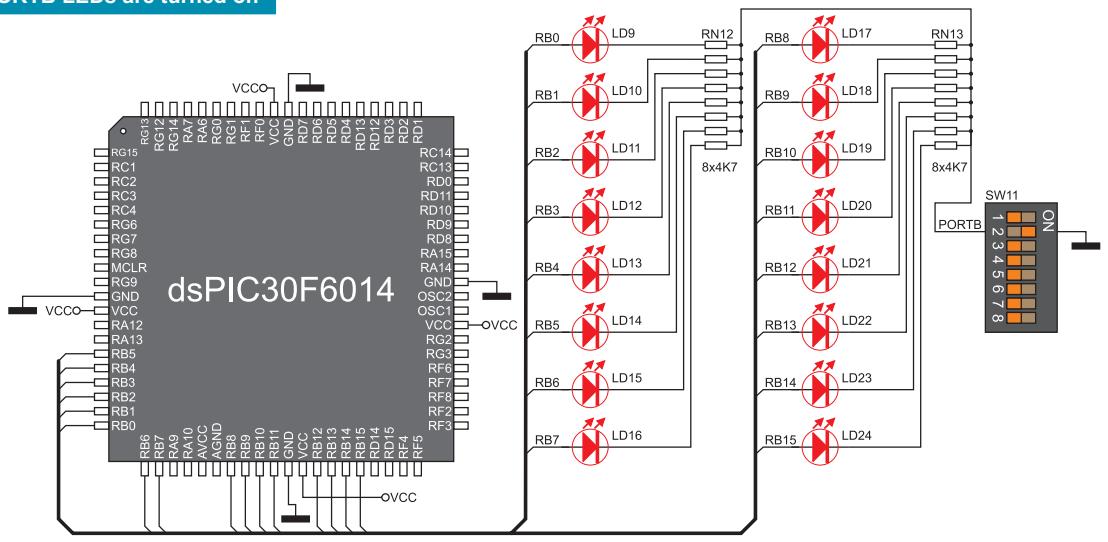


Figure 15-2: LEDs and port PORTB connection schematic

16.0. Push Buttons

The logic state of all microcontroller input pins may be changed by means of push buttons. Jumper J13 is used to determine the logic state to be applied to the desired microcontroller pin by pressing appropriate push button. The function of the protective resistor is to limit the maximum current, thus preventing the development system and peripheral modules from being damaged in case a short circuit occurs. If needed, advanced users may shorten such resistor using jumper J12. Right next to the push buttons, there is a RESET button which is not directly connected to the microcontroller pin. The reset signal is generated by the programmer.

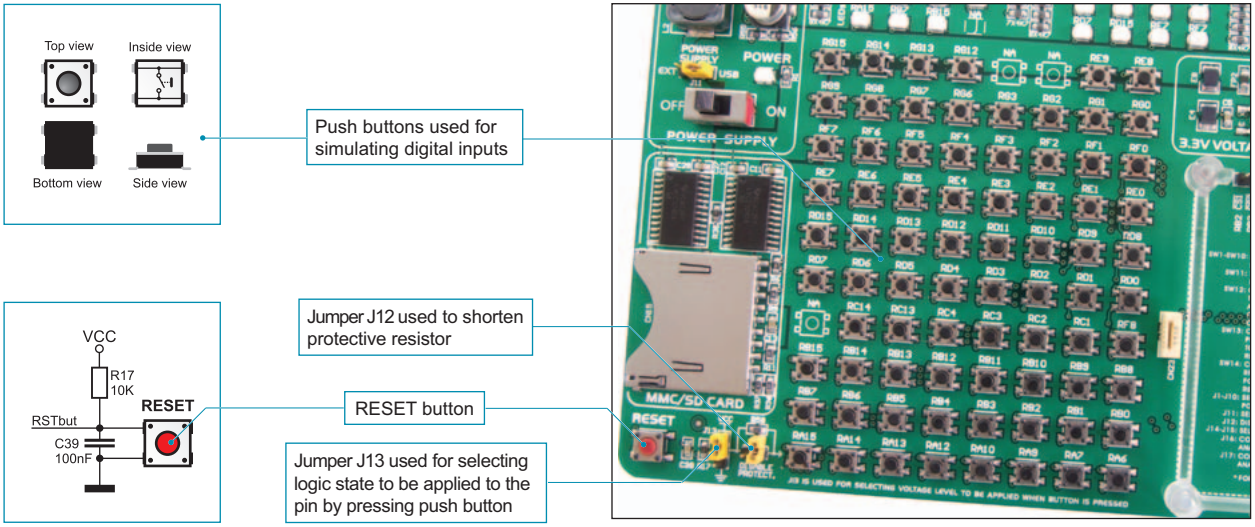


Figure 16-1: Push buttons

By pressing any push button when jumper J13 is in the VCC position, a logic one (5V) will be applied to the appropriate microcontroller pin, as shown in Figure 16-2.

By pressing a push button, the appropriate pin will be driven high (1)

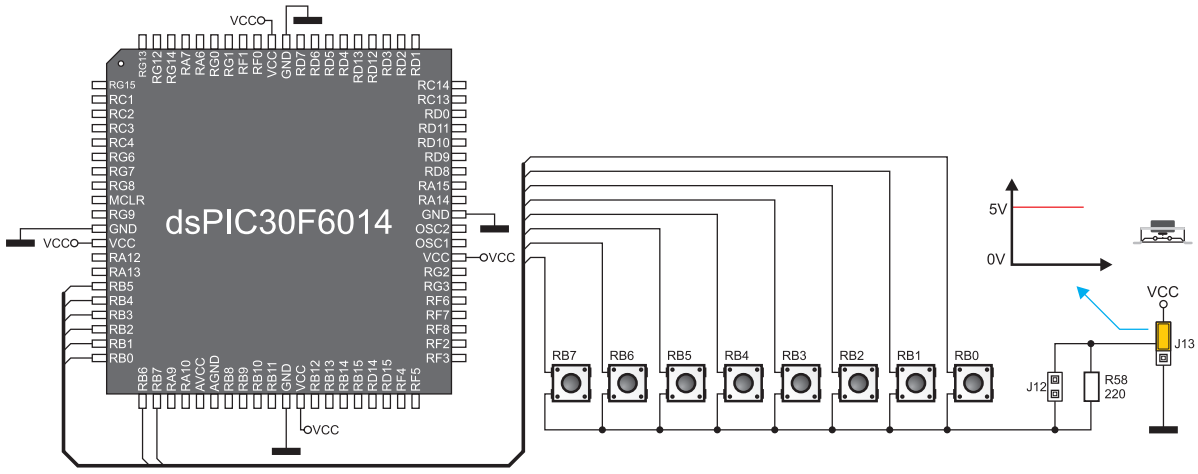


Figure 16-2: Push buttons and port PORTB connection schematic

page

17.0. 2x16 LCD

The *BIGdsPIC6* development system provides an on-board connector for the alphanumeric 2x16 LCD. This connector is linked to the microcontroller via ports PORTB and PORTD. Potentiometer P1 is used to adjust display contrast. Switch LCD-GLCD on the DIP switch SW11 is used to turn the display backlight on/off.
Communication between the LCD and the microcontroller is performed in a 4-bit mode. Alphanumeric digits are displayed in two lines each containing up to 16 characters of 7x5 pixels.

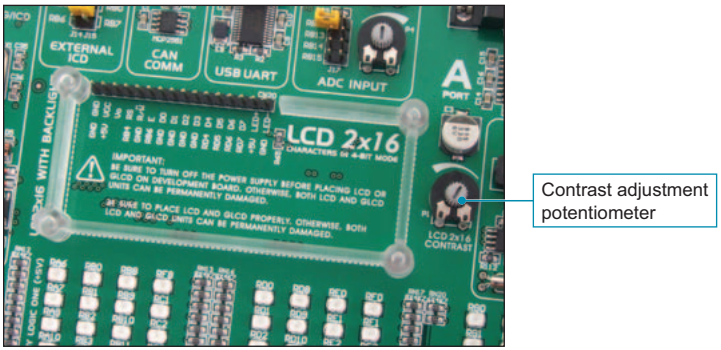


Figure 17-1. Alphanumeric LCD connector

Figure 17-2: Alphanumeric 2x16 LCD

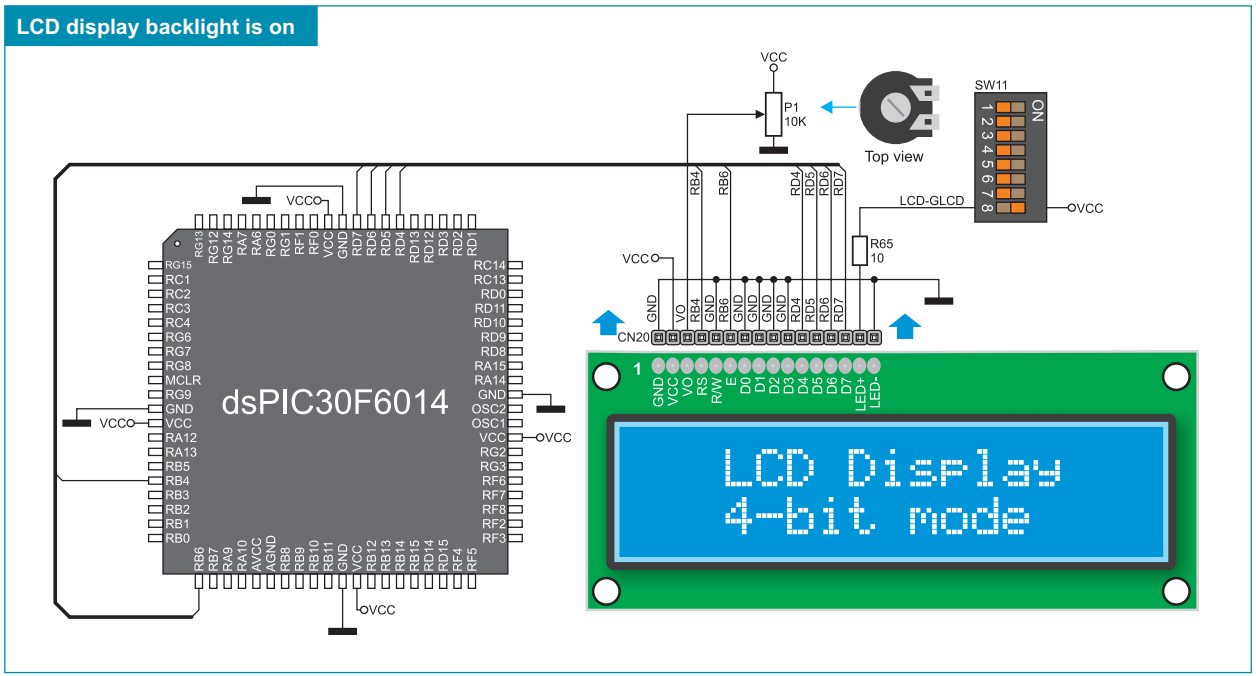


Figure 17-3: Alphanumeric 2x16 LCD connection schematic

18.0. 128x64 Graphic LCD

128x64 graphic LCD (GLCD) is connected to the microcontroller via PORTB and PORTD ports and enables graphic content to be displayed. It has the screen resolution of 128x64 pixels, which allows diagrams, tables and other graphic content to be displayed. As both ports are also used for the operation of 2x16 LCD, the displays cannot be used simultaneously. Potentiometer P2 is used for the GLCD display contrast adjustment. Switch 8 (LCD-GLCD) on the DIP switch SW11 is used to turn the display backlight on/off.

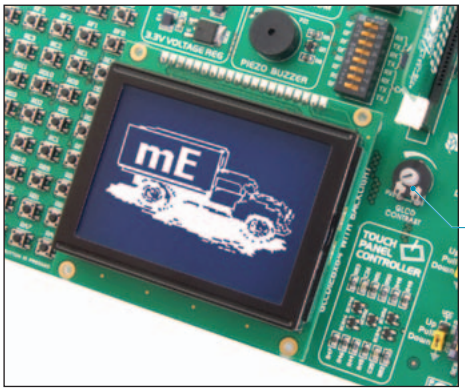


Figure 18-1: GLCD

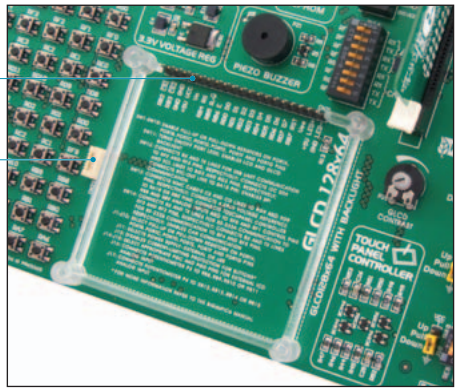


Figure 18-2: GLCD connector

GLCD display backlight is on

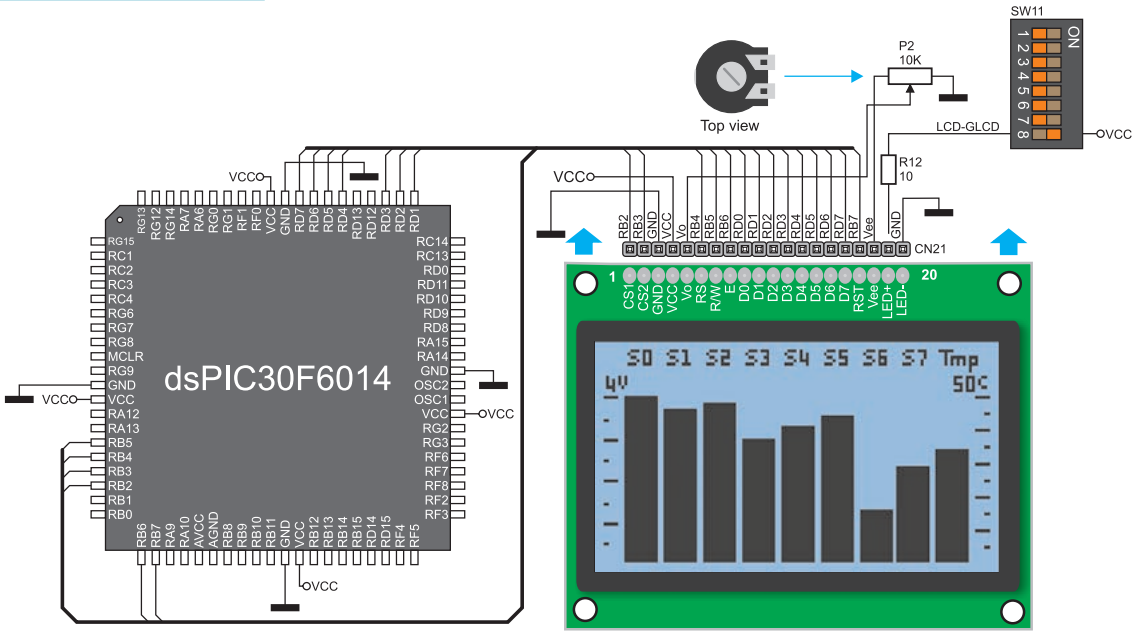


Figure 18-3: GLCD connection schematic

page

19.0. Touch Panel

The touch panel is a thin, self-adhesive, transparent, touch-sensitive panel. It is placed over a GLCD display. Its main function is to register pressure at some specific display point and to forward its coordinates in the form of analog voltage to the microcontroller. Switches 5, 6, 7 and 8 on the DIP switch SW13 are used to connect the microcontroller and touch panel.

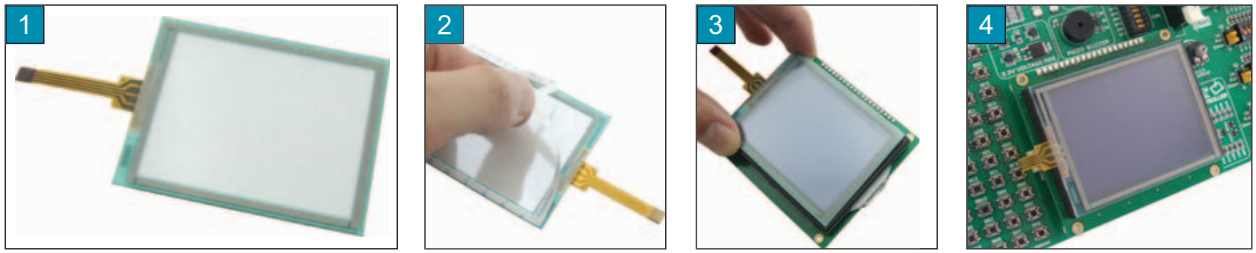


Figure 19-1: Placing touch panel over a GLCD

Figure 19-1 shows how to place a touch panel over a GLCD display. Make sure that the flat cable is to the left of the GLCD, as shown in Figure 4.

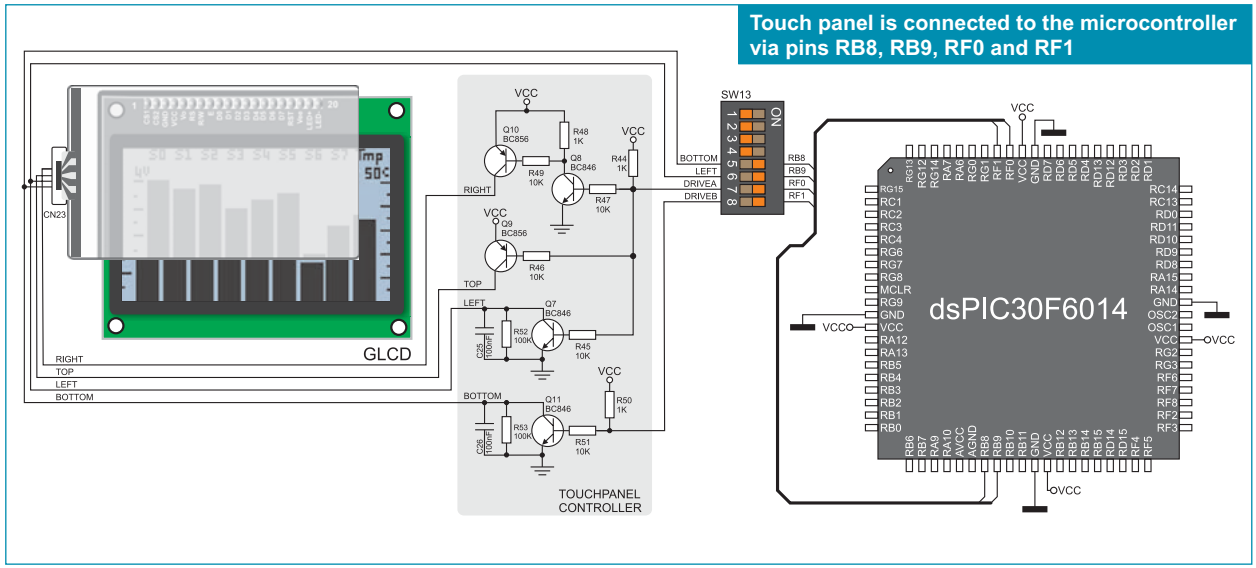


Figure 19-2: Touch panel connection schematic

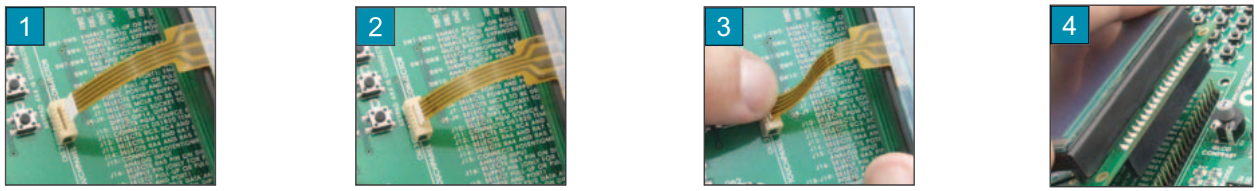


Figure 19-3: Connecting touch panel

Figure 19-3 shows in detail how to connect a touch panel to the microcontroller. Bring the end of the flat cable close to the CN23 connector (Figure 1). Plug the cable into the connector (Figure 2) and press it easily so as to fully fit the connector (Figure 3). Now, a GLCD can be plugged into the appropriate connector (Figure 4).

NOTE: LEDs and pull-up/pull-down resistors on ports PORTB and PORTF must be off when using a touch panel.

20.0. Input/Output Ports

Along the right side of the development system, there are ten 10-pin connectors connected to the microcontroller's I/O ports. Pins RB0, RB1, RB6 and RB7 are used for programming and therefore are not directly connected to the appropriate 10-pin connector, but via the programmer's multiplexer. DIP switches SW1-SW10 enable each connector pin to be connected to one pull-up/pull-down resistor. Whether pins of some port are to be connected to a pull-up or a pull-down resistor depends on the position of jumpers J1-J10.

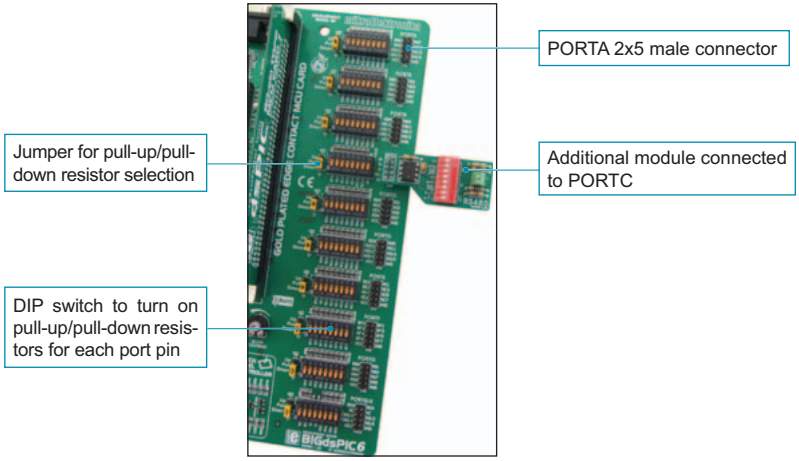

PORTA 2x5 male connector
Jumper for pull-up/pull-down resistor selection
Additional module connected to PORTC
DIP switch to turn on pull-up/pull-down resistors for each port pin

Figure 20-1: I/O ports

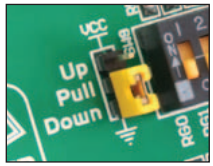


Figure 20-2: J9 in pull-down position

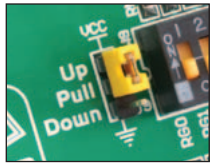


Figure 20-3: J9 in pull-up position

Port PORTB pins are connected to pull-down resistors

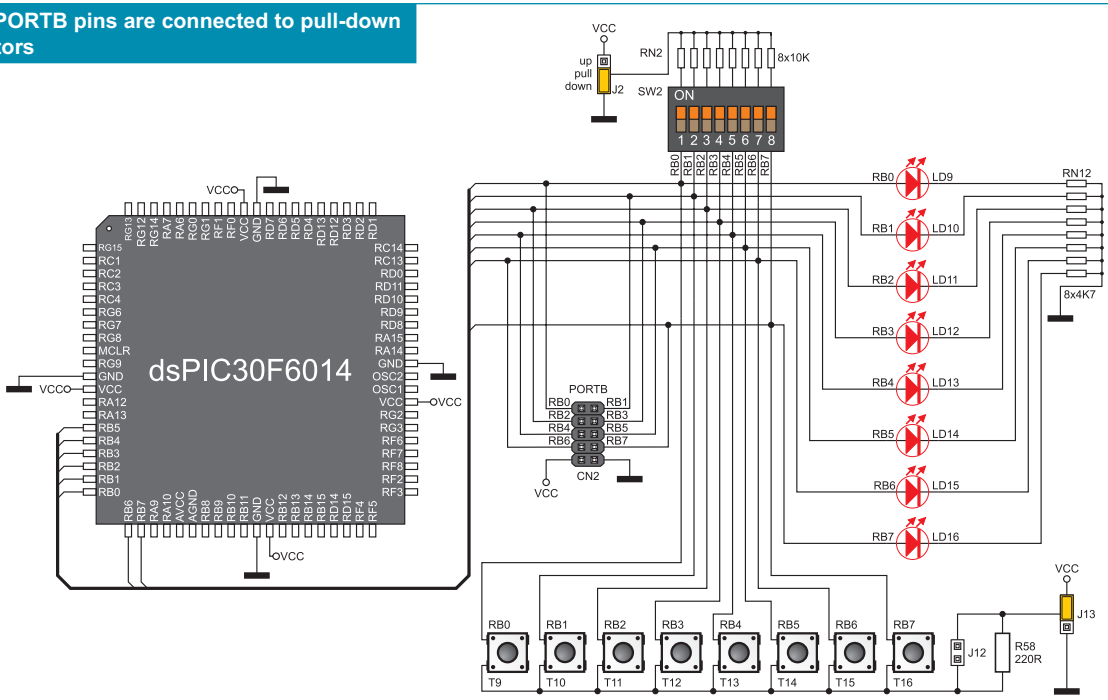


Figure 20-4: Port PORTB connection schematic

Pull-up/pull-down resistors enable you to set the logic level on all microcontroller's input pins when they are in idle state. Such level depends on the position of the pull-up/pull-down jumper. The RD8 pin with the relevant DIP switch SW6, jumper J6 and RD8 push button with jumper J13 are used here for the purpose of explaining the performance of pull-up/pull-down resistors. The principle of their operation is the same as for all other microcontroller pins.

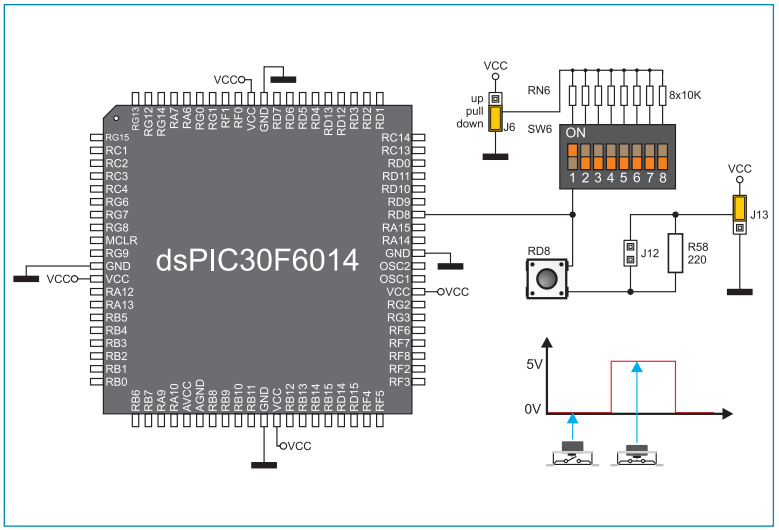


Figure 20-5: Jumper J6 in pull-down position and jumper J13 in pull-up position

In order to enable the PORTD port pins to be connected to pull-down resistors, it is necessary to place jumper J6 in the *Down* position first. This enables any PORTD port pin to be supplied with a logic zero (0V) in idle state over jumper J6 and 8x10k resistor network. To provide the RD8 pin with such signal, it is necessary to set switch RD8 on the DIP switch SW6 to the ON position.

As a result, every time you press the RD8 push button, a logic one (VCC voltage) will appear on the RD8 pin, provided that jumper J13 is placed in the VCC position.

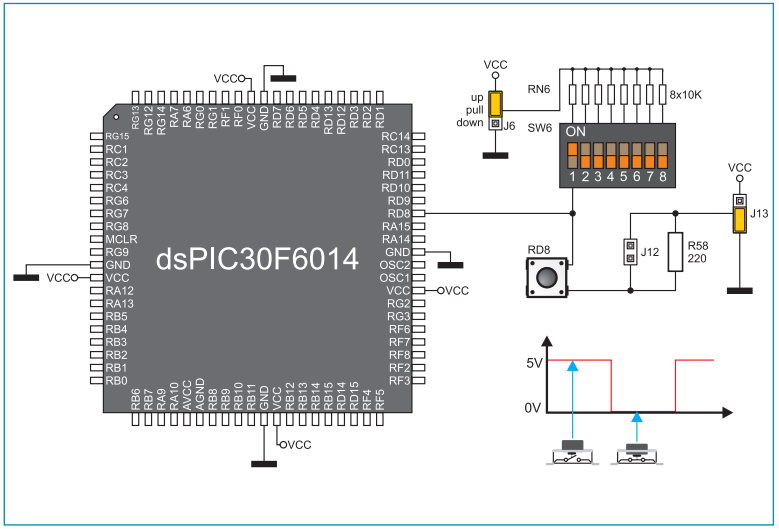


Figure 20-6: Jumper J6 in pull-up and jumper J13 in pull-down position

In order to enable port PORTD pins to be connected to pull-up resistors and the port input pins to be supplied with a logic zero (0), it is necessary to place jumper J6 in the *Up* position and jumper J13 in the GND position.
This enables any port PORTD input pin, when it is in idle state, to be driven high (5V) over the 10k resistor.

As a result, every time you press the RD8 push button, a logic zero (0V) will appear on the RD8 pin, provided that the RD8 switch is set to the ON position.

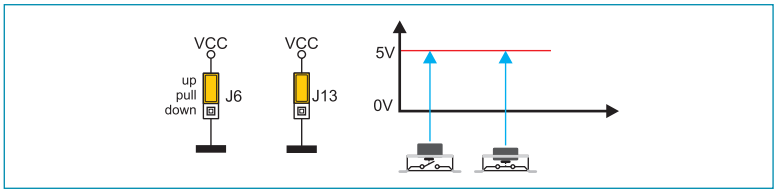


Figure 20-7: Jumpers J6 and J13 in the same positions

In case that jumpers J6 and J13 have the same logic state, pressure on any button will not cause input pins to change their logic state.

DISCLAIMER

All the products owned by MikroElektronika are protected by copyright law and international copyright treaty. Therefore, this manual is to be treated as any other copyright material. No part of this manual, including product and software described herein, may be reproduced, stored in a retrieval system, translated or transmitted in any form or by any means, without the prior written permission of MikroElektronika. The manual PDF edition can be printed for private or local use, but not for distribution. Any modification of this manual is prohibited.

MikroElektronika provides this manual 'as is' without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties or conditions of merchantability or fitness for a particular purpose.

MikroElektronika shall assume no responsibility or liability for any errors, omissions and inaccuracies that may appear in this manual. In no event shall MikroElektronika, its directors, officers, employees or distributors be liable for any indirect, specific, incidental or consequential damages (including damages for loss of business profits and business information, business interruption or any other pecuniary loss) arising out of the use of this manual or product, even if MikroElektronika has been advised of the possibility of such damages. MikroElektronika reserves the right to change information contained in this manual at any time without prior notice, if necessary.

HIGH RISK ACTIVITIES

The products of MikroElektronika are not fault – tolerant nor designed, manufactured or intended for use or resale as on – line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of Software could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities'). MikroElektronika and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

TRADEMARKS

The Mikroelektronika name and logo, the Mikroelektronika logo, mikroC, mikroC PRO, mikroBasic, mikroBasic PRO, mikroPascal, mikroPascal PRO, AVRflash, PICflash, dsPICprog, 18FJprog, PSOCprog, AVRprog, 8051prog, ARMflash, EasyPIC5, EasyPIC6, BigPIC5, BigPIC6, dsPIC PRO4, Easy8051B, EasyARM, EasyAVR5, EasyAVR6, BigAVR2, EasydsPIC4A, EasyPSoC4, EasyVR Stamp LV18FJ, LV24-33A, LV32MX, PIC32MX4 MultiMedia Board, PICPLC16, PICPLC8 PICPLC4, SmartGSM/GPRS, UNI-DS are trademarks of Mikroelektronika. All other trademarks mentioned herein are property of their respective companies.

All other product and corporate names appearing in this manual may or may not be registered trademarks or copyrights of their respective companies, and are only used for identification or explanation and to the owners' benefit, with no intent to infringe.

©Mikroelektronika™, 2010, All Rights Reserved.

If you want to learn more about our products, please visit our website at www.mikroe.com

If you are experiencing some problems with any of our products or just need additional information, please place your ticket at www.mikroe.com/en/support

If you have any questions, comments or business proposals, do not hesitate to contact us at office@mikroe.com

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

mikroElektronika:
MIKROE-504